Undecidability of the Word Problem

Fanxin Wu

April 11, 2022

UIC Distinguished Lecture Series

April 13-15, The University of Illinois at Chicago

Featuring three lectures by Hugh Woodin

Public Lecture,
4pm April 13, LC D5

Infinite Necessity

Funded by NSF
grant no. 1945251

Additional talks by Juliette Kennedy, Paul Larson, and Caroline Terry.

Presentation of a group

Let A be any set. The free group over A is the set of reduced word in A under concatenation; a word is reduced if there is no appearance of $a^{-1}a$ or aa^{-1} for $a \in A$.

Presentation of a group

Let A be any set. The free group over A is the set of reduced word in A under concatenation; a word is reduced if there is no appearance of $a^{-1}a$ or aa^{-1} for $a \in A$.

Definition 1

Let R be a subset of $F(A)$. $\langle A \mid R \rangle$ is the quotient of $F(A)$ by the normal subgroup generated by R. If $G \simeq \langle A \mid R \rangle$ where both A, R are finite, then G is *finitely presented*.

Presentation of a group

Examples:

1. $\langle a, b \mid aba^{-1}b^{-1} \rangle$ is the free abelian group in two generators. We also write $\langle a, b \mid aba^{-1}b^{-1} = 1 \rangle$ or $\langle a, b \mid ab = ba \rangle$.

2. If $G = \langle A \mid R \rangle$ and $H = \langle A' \mid R' \rangle$, then $G * H := \langle A, A' \mid R, R' \rangle$ is called the *free product* of G and H. For example, $\langle a, b \mid a^2, b^3 \rangle$ is the free product of $\mathbb{Z}/2\mathbb{Z}$ and $\mathbb{Z}/3\mathbb{Z}$. It is also isomorphic to $\mathsf{PSL}(2, \mathbb{Z})$. Every element in $G * H$ is equivalent to a unique reduced word $g_1 h_1 \cdots g_n h_n$.

The word problem

If $G = \langle A \mid R \rangle$ where A is finite, and there exists an algorithm that decides whether a given word in A represents the trivial element in G, then the word problem for G is *decidable*.

The word problem

If $G = \langle A \mid R \rangle$ where A is finite, and there exists an algorithm that decides whether a given word in A represents the trivial element in G, then the word problem for G is *decidable*.

Facts:

1. This does not depend on the generating set A.

2. The word problem for free groups is decidable since every word is equivalent to a unique reduced word.

3. If the words in R "do not overlap too much", then the word problem is decidable using Dehn's algorithm.

The word problem

If $G = \langle A \mid R \rangle$ where A is finite, and there exists an algorithm that decides whether a given word in A represents the trivial element in G, then the word problem for G is *decidable*.

Facts:

1. This does not depend on the generating set A.

2. The word problem for free groups is decidable since every word is equivalent to a unique reduced word.

3. If the words in R "do not overlap too much", then the word problem is decidable using Dehn's algorithm.

4. There exists a finitely presented group whose word problem is undecidable.

The word problem

If $G = \langle A \mid R \rangle$ where A is finite, and there exists an algorithm that decides whether a given word in A represents the trivial element in G, then the word problem for G is *decidable*.

Facts:

1. This does not depend on the generating set A.

2. The word problem for free groups is decidable since every word is equivalent to a unique reduced word.

3. If the words in R "do not overlap too much", then the word problem is decidable using Dehn's algorithm.

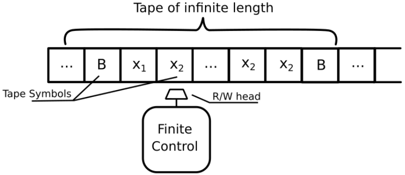
4. There exists a finitely presented group whose word problem is undecidable.

We need to make the notion of "algorithm" more precise.

Turing machine

Very roughly speaking, a *Turing machine* is a computer with infinite memory and power. Slightly more mathematically, a Turing machine consists of an infinitely long *tape* divided into cells, a *head* that can read or write symbols on the tape, along with:

1. a finite set of possible *states*,
2. a finite set of allowed *symbols*,
3. a finite set of *instructions* that specify whether to halt/write a symbol/move the head depending on the current state and the current symbol the head is reading.



Computability

Given a Turing machine and a finite *input* (only finitely many cells are not blank), we can run the machine according to the instructions. It may or may not run forever. In case it halts, the final configuration is called the *output*.

It is harmless to assume the only allowed symbols on the tape are $0, 1$ and B (blank). Using binary expansion, an input/output can be regarded as a natural number. This way a Turing machine can be viewed as computing a partial function on natural numbers.

Computability

Given a Turing machine and a finite *input* (only finitely many cells are not blank), we can run the machine according to the instructions. It may or may not run forever. In case it halts, the final configuration is called the *output*.

It is harmless to assume the only allowed symbols on the tape are $0, 1$ and B (blank). Using binary expansion, an input/output can be regarded as a natural number. This way a Turing machine can be viewed as computing a partial function on natural numbers.

Definition 2

A function $f : \mathbb{N} \to \mathbb{N}$ is called *computable* if there exists a Turing machine that always halts and outputs $f(n)$ (given input n).

Computability

Definition 3

The class of *partial recursive functions* is the smallest class of functions $f : \mathbb{N}^k \to \mathbb{N}$ for some $k \geq 1$ that contains:

1. constant functions $f(x_1, ..., x_k) = c$,
2. successor function $S(x) = x + 1$,
3. projections $p(x_1, ..., x_k) = x_i$,

and closed under

1. composition,
2. primitive recursion, $f(x + 1) = g(x, f(x))$,
3. minimizing operation, $\mu_f(x) =$ the least y s.t. $f(x, y) = 0$.

If moreover f is total, then it is called recursive.

Computability

Theorem

A function $f : \mathbb{N} \to \mathbb{N}$ is computable iff it is recursive.

Computability

Theorem

A function $f : \mathbb{N} \to \mathbb{N}$ is computable iff it is recursive.

Thesis (Church-Turing)

1. Any two reasonable definitions of "computable functions" are equivalent.
2. Moreover, they correctly capture the notion of algorithm.

Computability

Theorem

A function $f : \mathbb{N} \to \mathbb{N}$ *is computable iff it is recursive.*

Thesis (Church-Turing)

1. Any two reasonable definitions of "computable functions" are equivalent.
2. Moreover, they correctly capture the notion of algorithm.

Consequences:

1. Any reasonable variants of Turing machine are equivalent.

2. If a function is obviously computable by a real-life computer, then it is computable.

Formulating the word problem

Let $G = \langle A \mid R \rangle$ be f.g (or more generally countably generated). WLOG $A \subseteq \mathbb{N}$, so a word in A is an element of $\mathbb{N}^{<\omega} := \bigcup_{k \geq 1} \mathbb{N}^k$; we want to code it into a natural number.

Formulating the word problem

Let $G = \langle A \mid R \rangle$ be f.g (or more generally countably generated). WLOG $A \subseteq \mathbb{N}$, so a word in A is an element of $\mathbb{N}^{<\omega} := \bigcup_{k \geq 1} \mathbb{N}^k$; we want to code it into a natural number.

Any reasonable bijections between $\mathbb{N}^{<\omega}$ and $\mathbb{N}$ are equivalent, e.g., lexicographical ordering, prime factorization, Chinese Remainder Theorem...

Formulating the word problem

Let $G = \langle A \mid R \rangle$ be f.g (or more generally countably generated). WLOG $A \subseteq \mathbb{N}$, so a word in A is an element of $\mathbb{N}^{<\omega} := \bigcup_{k \geq 1} \mathbb{N}^k$; we want to code it into a natural number.

Any reasonable bijections between $\mathbb{N}^{<\omega}$ and $\mathbb{N}$ are equivalent, e.g., lexicographical ordering, prime factorization, Chinese Remainder Theorem...

We say that the word problem for G is decidable if the normal subgroup generated by R is recursive in $F(A)$, i.e., there is a computable $f : F(A) \to \{0, 1\}$ such that

$$f(w) = \begin{cases} 1 & w \text{ represents identity in } G \\ 0 & \text{otherwise} \end{cases}$$

More computability

Definition 4

$X \subseteq \mathbb{N}$ is called *recursive* if the characteristic function on X is computable. X is *recursively enumerable* if there exists a non-stop Turing machine that lists the members of X (in any order!), or equivalently X is the range of a computable function.

More computability

Definition 4

$X \subseteq \mathbb{N}$ is called *recursive* if the characteristic function on X is computable. X is *recursively enumerable* if there exists a non-stop Turing machine that lists the members of X (in any order!), or equivalently X is the range of a computable function.

Examples:

1. A recursive set is recursively enumerable.

2. If both X and $\mathbb{N} \setminus X$ are r.e., then X is recursive.

3. There exists a r.e. set that is not recursive, such as $\{n \mid \text{the } n\text{-th Turing machine halts with empty input}\}$.

4. Consider a f.g. group $G = \langle A \mid R \rangle$. If the word problem for G is decidable, then the normal subgroup $\langle R \rangle^{F(A)}$ generated by R in $F(A)$ is recursive, and by replacing R with $\langle R \rangle^{F(A)}$ we may assume R is recursive. Therefore when discussing word problem we focus on *recursively presented groups*. If R is recursive then $\langle R \rangle^{F(A)}$ is always r.e.

5. If G has a r.e. presentation, then it has a recursive presentation.

More computability

$\{(a_1, ..., a_n) \in \mathbb{Z}^n \mid f(a_1, ..., a_n) = 0\}$ is recursive for any $f(x_1, ..., x_n) \in \mathbb{Z}[x_1, ..., x_n]$.

$\{(a_1, ..., a_n) \in \mathbb{Z}^n \mid \exists b_1 \cdots b_m \ f(a_1, ..., a_n, b_1, ..., b_m) = 0\}$ is r.e. for any $f(x_1, ..., x_n, y_1, ..., y_m) \in \mathbb{Z}[x_1, ..., x_n, y_1, ..., y_m]$. Such a set is also called *Diophantine*.

More computability

$\{(a_1, ..., a_n) \in \mathbb{Z}^n \mid f(a_1, ..., a_n) = 0\}$ is recursive for any $f(x_1, ..., x_n) \in \mathbb{Z}[x_1, ..., x_n]$.

$\{(a_1, ..., a_n) \in \mathbb{Z}^n \mid \exists b_1 \cdots b_m \; f(a_1, ..., a_n, b_1, ..., b_m) = 0\}$ is r.e. for any $f(x_1, ..., x_n, y_1, ..., y_m) \in \mathbb{Z}[x_1, ..., x_n, y_1, ..., y_m]$. Such a set is also called *Diophantine*.

Theorem (Matiyasevich–Robinson–Davis–Putnam)

Every r.e. set is Diophantine.

More computability

$\{(a_1, ..., a_n) \in \mathbb{Z}^n \mid f(a_1, ..., a_n) = 0\}$ is recursive for any $f(x_1, ..., x_n) \in \mathbb{Z}[x_1, ..., x_n]$.

$\{(a_1, ..., a_n) \in \mathbb{Z}^n \mid \exists b_1 \cdots b_m \ f(a_1, ..., a_n, b_1, ..., b_m) = 0\}$ is r.e. for any $f(x_1, ..., x_n, y_1, ..., y_m) \in \mathbb{Z}[x_1, ..., x_n, y_1, ..., y_m]$. Such a set is also called *Diophantine*.

Theorem (Matiyasevich–Robinson–Davis–Putnam)

Every r.e. set is Diophantine.

Corollary (Hilbert's tenth problem)

There does not exist general algorithm that decides whether a polynomial has integer solution.

HNN extension

Suppose G has presentation $\langle A \mid R \rangle$, H, H' are subgroups of G and $\phi : H \to H'$ is an isomorphism.

Definition 5

$G^* := \langle A \cup \{t\} \mid R \cup \{t^{-1}ht = \phi(h) \mid h \in H\} \rangle$ is the *HNN extension* by the stable letter t with respect to ϕ. We also write $\langle G, t \mid t^{-1}ht = \phi(h), h \in H \rangle$.

More generally we may consider a collection of isomorphisms $\phi_i : H_i \to H_i'$ and stable letters t_i.

HNN extension

Suppose G has presentation $\langle A \mid R \rangle$, H, H' are subgroups of G and $\phi : H \to H'$ is an isomorphism.

Definition 5

$G^* := \langle A \cup \{t\} \mid R \cup \{t^{-1}ht = \phi(h) \mid h \in H\} \rangle$ is the *HNN extension* by the stable letter t with respect to ϕ. We also write $\langle G, t \mid t^{-1}ht = \phi(h), h \in H \rangle$.

More generally we may consider a collection of isomorphisms $\phi_i : H_i \to H'_i$ and stable letters t_i.

Theorem 6

1. The natural map from G to G^* is injective.
2. (Britton's lemma) If the sequence $g_0 t^{\epsilon_1} g_1 \cdots t^{\epsilon_n} g_n$ represents identity in G^*, where $g_k \in G$, $\epsilon_k = \pm 1$ and $n \geq 1$, then there is an appearance of $t^{-1}ht$, $h \in H$ or $th't^{-1}$, $h' \in H'$.

HNN extension

Consequences:

1. G^* is a supergroup of G where H, H' are conjugate.
2. If G has decidable word problem then so does G^*.

HNN extension

Consequences:

1. G^* is a supergroup of G where H, H' are conjugate.
2. If G has decidable word problem then so does G^*.

Example:

$H = H'$ and ϕ is identity, so $G^* = \langle G, t \mid t^{-1}ht = h, h \in H \rangle$. We claim that for $g \in G$, $t^{-1}gt = g$ iff $g \in H$. Indeed, if $g \notin H$ then $t^{-1}gtg^{-1}$ contains no appearance of $t^{-1}ht$ or tht^{-1}.

Benign subgroup

Call a f.g. group G Higman if it can be embedded in a f.p. group.

Definition 7

A subgroup H of a f.g. group G is called *benign* in G if the group $G_H := \{G, t \mid t^{-1}ht = h, h \in H\}$ is Higman.

Note that H is not assumed to be f.g.

Benign subgroup

Call a f.g. group G Higman if it can be embedded in a f.p. group.

Definition 7

A subgroup H of a f.g. group G is called *benign* in G if the group $G_H := \{G, t \mid t^{-1}ht = h, h \in H\}$ is Higman.

Note that H is not assumed to be f.g.

Facts:

1. If G is f.p., then any f.g. subgroup H is benign. More generally this is true for G Higman: if $G \subseteq L$ where L is f.p. then $G_H \subseteq L_H$.

2. If we have $H \subseteq G \subseteq L$ where G, L are f.g., and H is benign in L, then H is benign in G, since $G_H \subseteq L_H$.

3. If $H, K \subseteq G$ are benign then so are $H \cap K$ and $Gp\{H, K\}$.

Benign subgroup

Theorem (Higman's Embedding Theorem)

A f.g. group is Higman iff it has a recursive presentation $\langle A \mid R \rangle$.

One direction is simple: Suppose $G = \langle A \mid R \rangle$ is embedded in a f.p. group $\langle A' \mid R' \rangle$; wlog assume $A \subseteq A'$. Then $\langle R \rangle^{F(A)} = \langle R' \rangle^{F(A')} \cap A$ is r.e.

Granted Higman's theorem, $H \subseteq G$ is benign roughly means H is a recursive subset of G. Higman's theorem easily implies undecidability of word problem.

The principal lemma

Lemma 8

Let $S \subseteq \mathbb{Z}$ be recursively enumerable. Then the subgroup $Gp\{a^z b c^z \mid z \in S\}$ is benign in the free group $\langle a, b, c \rangle$.

The principal lemma

Lemma 8

Let $S \subseteq \mathbb{Z}$ be recursively enumerable. Then the subgroup $Gp\{a^z b c^z \mid z \in S\}$ is benign in the free group $\langle a, b, c \rangle$.

Fact: $Gp\{a^z b c^z \mid z \in \mathbb{Z}\}$ is freely generated by the indicated generators.

Proof of undecidability assuming the principal lemma.

Note that if $G \subseteq L$ are both f.g. and the word problem for L is decidable, then the same holds for G.

Let S be a r.e. set that is not recursive. Let $G = \langle a, b, c \rangle$ and $B = Gp\{a^z bc^z \mid z \in S\}$. $G_B = \langle a, b, c, t \mid t^{-1}a^z bc^z t = a^z bc^z \rangle$ can be embedded into a f.p. group L by the principal lemma. $t^{-1}a^z bc^z t = a^z bc^z$ iff $a^z bc^z \in B$, and this happens iff $z \in S$ by freeness.

If there were an algorithm that solves the word problem of G_B, we could use it to determine whether a given integer z is in S. Therefore G_B's word problem is undecidable, and so is L. $\square$

Step 1: elementary formulas

By MRDP Theorem, we may assume S is Diophantine, so there is some polynomial $f(z_0, z_1, ..., z_t)$ with integer coefficients s.t.

$z_0 \in S \Leftrightarrow \exists z_1 \cdots z_t \ f(z_0, z_1, ..., z_t) = 0$

Step 1: elementary formulas

By MRDP Theorem, we may assume S is Diophantine, so there is some polynomial $f(z_0, z_1, ..., z_t)$ with integer coefficients s.t.

$z_0 \in S \Leftrightarrow \exists z_1 \cdots z_t \; f(z_0, z_1, ..., z_t) = 0$

We can convert this to

$z_0 \in S \Leftrightarrow \exists z_1 \cdots z_m \phi(z_0, z_1, ..., z_m)$

where $\phi = \bigwedge_p \phi_p$ is the conjunction of some elementary formulas of one of the forms

$z_i = c$ (c an integer)

$z_i = z_j$

$z_i + z_j = z_k$ (i, j, k distinct)

$z_l = z_i \cdot z_j$ ($0 < l < i < j \leq m$)

Step 1: elementary formulas

Example:

$$
\begin{aligned}
&\exists z_1 \; z_0^2 + z_1 + 1 = 0 \\
\Leftrightarrow &\exists z_1 \exists z_2 \; (z_2 = z_0^2) \wedge (z_2 + z_1 + 1 = 0) \\
\Leftrightarrow &\exists z_1 \exists z_2 \exists z_3 \; (z_2 = z_0^2) \wedge (z_3 = z_2 + z_1) \wedge (z_3 = -1) \\
\Leftrightarrow &\exists z_1 \exists z_2 \exists z_3 \exists z_4 \exists z_5 \; (z_2 = z_4 \cdot z_5) \wedge (z_4 = z_0) \wedge (z_5 = z_0) \\
&\qquad\qquad\qquad\qquad \wedge (z_3 = z_2 + z_1) \wedge (z_3 = -1)
\end{aligned}
$$

Step 2: reducing to elementary formulas

Goal: Let $S = \{z_0 \mid \exists z_1 \cdots z_m \phi(z_0, z_1, ..., z_m)\}$. Then the subgroup $B = Gp\{a_0^{z_0} b_0 c_0^{z_0} \mid z_0 \in S\}$ is benign in the free group $\langle a_0, b_0, c_0 \rangle$.

Let $F = \langle a_0, b_0, c_0, ..., a_m, b_m, c_m \rangle$. For each tuple $(z_0, ..., z_m) \in \mathbb{Z}^{m+1}$, define $w_{(z_0,...,z_m)}$ as follows:

$$c_m^{-z_m} b_m^{-1} a_m^{-z_m} \cdots c_1^{-z_1} b_1^{-1} a_1^{-z_1} a_0^{z_0} b_0 c_0^{z_0} a_1^{z_1} b_1 c_1^{z_1} \cdots a_m^{z_m} b_m c_m^{z_m}$$

Step 2: reducing to elementary formulas

Goal: Let $S = \{z_0 \mid \exists z_1 \cdots z_m \phi(z_0, z_1, ..., z_m)\}$. Then the subgroup $B = Gp\{a_0^{z_0} b_0 c_0^{z_0} \mid z_0 \in S\}$ is benign in the free group $\langle a_0, b_0, c_0 \rangle$.

Let $F = \langle a_0, b_0, c_0, ..., a_m, b_m, c_m \rangle$. For each tuple $(z_0, ..., z_m) \in \mathbb{Z}^{m+1}$, define $w_{(z_0,...,z_m)}$ as follows:

$$c_m^{-z_m} b_m^{-1} a_m^{-z_m} \cdots c_1^{-z_1} b_1^{-1} a_1^{-z_1} a_0^{z_0} b_0 c_0^{z_0} a_1^{z_1} b_1 c_1^{z_1} \cdots a_m^{z_m} b_m c_m^{z_m}$$

Fact: $\{w_{(z_0,...,z_m)} \mid (z_0, ..., z_m) \in \mathbb{Z}^{m+1}\}$ freely generate a subgroup $A \subseteq F$.

Step 2: reducing to elementary formulas

Let $A_\phi = Gp\{w_{(z_0,...,z_m)} \mid \phi(z_0, ..., z_m) \text{ is true}\}$;

$A_i^c = Gp\{w_{(z_0,...,z_m)} \mid z_i = c\}$;

$A_{i,j}^= = Gp\{w_{(z_0,...,z_m)} \mid z_i = z_j\}$, etc.

Notice that $F(X) \cap F(Y) = F(X \cap Y)$. So A_ϕ is the intersection of all the A_i^c, $A_{i,j}^=$, etc.

Step 2: reducing to elementary formulas

Let $A_\phi = Gp\{w_{(z_0,...,z_m)} \mid \phi(z_0, ..., z_m)$ is true$\}$;

$A_i^c = Gp\{w_{(z_0,...,z_m)} \mid z_i = c\}$;

$A_{i,j}^= = Gp\{w_{(z_0,...,z_m)} \mid z_i = z_j\}$, etc.

Notice that $F(X) \cap F(Y) = F(X \cap Y)$. So A_ϕ is the intersection of all the A_i^c, $A_{i,j}^=$, etc.

Recall that $B = Gp\{a_0^{z_0} b_0 c_0^{z_0} \mid z_0 \in S\}$.

$$\begin{aligned} & Gp\{A_\phi, a_1, b_1, c_1, ..., a_m, b_m, c_m\} \cap \{a_0, b_0, c_0\} \\ =& Gp\{B, a_1, b_1, c_1, ..., a_m, b_m, c_m\} \cap \{a_0, b_0, c_0\} \\ =& B \end{aligned}$$

Step 3: the HNN extension

As an example we show that $A_i^c = Gp\{w_{(z_0,...,z_m)} \mid z_i = c\}$ is benign in $F = \langle a_0, b_0, c_0, ..., a_m, b_m, c_m \rangle$. It suffices to construct a f.p. group $M \supseteq F$ and a f.g. $L \subseteq M$ s.t. $L \cap F = A_i^c$.

Step 3: the HNN extension

As an example we show that $A_i^c = Gp\{w_{(z_0,...,z_m)} \mid z_i = c\}$ is benign in $F = \langle a_0, b_0, c_0, ..., a_m, b_m, c_m\rangle$. It suffices to construct a f.p. group $M \supseteq F$ and a f.g. $L \subseteq M$ s.t. $L \cap F = A_i^c$.

For each i, $\{a_0, b_0, c_0, ..., a_i, a_ib_ic_i, c_i, ..., a_m, b_m, c_m\}$ freely generate F. Let M be the HNN extension of F by stable letters $t_i, 0 \leq i \leq m$ defined by

$t_i^{-1}b_it_i = a_ib_ic_i$, and

t_i commutes with all other generators of F.

Step 4: finishing the proof

Recall the definition of $w_{(z_0,\ldots,z_m)}$:

$c_m^{-z_m} b_m^{-1} a_m^{-z_m} \cdots c_1^{-z_1} b_1^{-1} a_1^{-z_1} a_0^{z_0} b_0 c_0^{z_0} a_1^{z_1} b_1 c_1^{z_1} \cdots a_m^{z_m} b_m c_m^{z_m}$

Note that $t_i^{-1} w_{(z_0,\ldots,z_m)} t_i = w_{(z_0,\ldots,z_i+1,\ldots,z_m)}$.

Step 4: finishing the proof

Recall the definition of $w_{(z_0,\dots,z_m)}$:

$c_m^{-z_m} b_m^{-1} a_m^{-z_m} \cdots c_1^{-z_1} b_1^{-1} a_1^{-z_1} a_0^{z_0} b_0 c_0^{z_0} a_1^{z_1} b_1 c_1^{z_1} \cdots a_m^{z_m} b_m c_m^{z_m}$

Note that $t_i^{-1} w_{(z_0,\dots,z_m)} t_i = w_{(z_0,\dots,z_i+1,\dots,z_m)}$.

So for each fixed i, $L := Gp\{A_i^c, t_j, j \neq i\}$ is generated by $w_{(0,\dots,c,\dots,0)}$ and $t_j, j \neq i$.

$L \cap F = A_i^c$ by Britton's Lemma.

Adian-Rabin Theorem

Theorem 9

If P is any property about finitely presented groups that is Markov*, then there does not exist an algorithm that decides whether a f.p. group has P from its presentation. P is Markov if:*

1. There is a f.p. group G_1 with P;

2. There is a f.p. group G_2 which cannot be embedded into any f.p. group with P.

Adian-Rabin Theorem

Theorem 9

If P is any property about finitely presented groups that is Markov, *then there does not exist an algorithm that decides whether a f.p. group has P from its presentation. P is Markov if:*
1. There is a f.p. group G_1 with P;
2. There is a f.p. group G_2 which cannot be embedded into any f.p. group with P.

Examples of Markov properties: being trivial, finite, free, abelian, torsion-free, simple, amenable, hyperbolic, property (T), etc.

Adian-Rabin Theorem

Proof idea: Let H be a f.p. group with undecidable word problem. Using HNN extension and free product with amalgamation, assign to each word w on the generators of H a presentation D_w in a computable way, so that D_w contains G_2 when $w \neq 1$ and D_w is trivial when $w = 1$. Then $D_w * G_1$ has property P iff $w = 1$ in H.

This "reduces" the word problem for H to the problem of deciding whether a f.p. group has property P. Since the former is undecidable, so is the latter.

Reference

N. Koblitz, B. Zilber, and Y.I. Manin.
A Course in Mathematical Logic for Mathematicians.
Graduate Texts in Mathematics. Springer New York, 2009.

R.C. Lyndon and P.E. Schupp.
Combinatorial Group Theory.
Classics in Mathematics. Springer Berlin Heidelberg, 2015.